INTEGRATION OF PLC WITH EPICS IOC FOR SUPERKEKB CONTROL SYSTEM

J. Odagiri, K. Akai, T. Ishibashi, K. Furukawa, H. Hisamatus, H. Kaji, T. Kobayashi, M. Mikawa, T. Mimashi, T. T. Nakamura, K. Nakanishi, M. Satoh, S. Terui, High Energy Acceterator Resarch Organization (KEK), Oho 1-1, Tsukuba, Ibarai, Japan
N. Yoshifuji, East Japan Institute of Technology Co. Ltd., Minatomachi 1-8-4, Tsuchiura, Ibaraki, Japan
T. Nakamura, M. Tanaka, K. Yoshii, Mitsubishi Electric System & Service Co. Ltd., Umezono 2-8-8, Tsukuba, Ibaraki, Japan
H. Deguchi, K. Hayashi, Mitsubishi Electric TOKKI Systems Corporation, Techno-parc 4-11, Sanda, Hyogo, Japan
T. Natsui, H. Shiratsu, Yokogawa Electric Corporation, Nakamachi 2-9-32, Musashino, Tokyo, Japan

Abstract

Recently, more and more PLCs are adopted for various frontend controls of accelerators. It is common to connect the PLCs with higher level frontend computers by the network. As a result, control logic becomes dispersed over separate layers, one of which is implemented by ladder programs for PLCs, and the other is implemented by higher level languages for frontend computers.

EPICS-based SuperKEKB [1] accelerator control system, however, take a different approach by using FA-M3 PLCs with its special CPU module, F3RP61, which adopted Linux as its OS and therefore can function as Input/output Controller (IOC. This consolidation of PLC and IOC enables higher level applications to directly reach every PLC placed at frontends by Channel Access (CA). In addition, most of control logic can be implemented by the IOC core program (iocCore) and/or EPICS sequencer to make the system more homogeneous. It results in easier development and maintenance of applications programs. This type of FA-M3-based IOCs are to be used to monitor and control many subsystems of SuperKEKB, such a Low Level RF (LLRF) system, a vacuum control system, a Personnel Protection System (PPS) and so on. This paper describes the FA-M3-based IOC and its applications to the SuperKEKB accelerator control system.

INTRODUCTION

The frontend hardware interfaces of the KEKB accelerator control system comprised VME singe board computers, which work as IOC, and non-intelligent interfaces such as CAMAC and GPIB.

Recently, however, PLCs are replacing those non-intelligent interfaces in favour of their rich functionality, such as high performance CPU, wide variety of I/O modules, capability of network connection, high resolution AD/DA modules, and so on. If the legacy interfaces are simply replaced with the PLCs, it result inhaving frontend controllers (PLCs) are placed under yet another frontend controllers (IOC). This unsolicited doubled layers, shown in (A) of Fig. 1, increases the effort to develop and maintain frontend control software

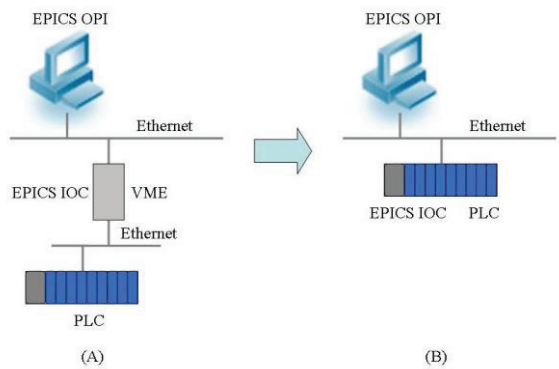

Figure 1: (A) shows traditional configuration of frontend control while (B) shows consolidation of IOC and PLC.

EPICS ON E3RP61

In order to solve the problem mentioned above, i.e., unsolicited doubled layers of frontend, we have adopted FA-M3 PLC with F3RP61 as IOC. The FA-M3-based IOC is shown in (B) of Fig. 1.

Features of F3RP61

F3RP61 CPU has the features as follows.

- Since F3RP61 executes Linux as its OS, it can execute complete IOC core program (iocCore) and related libraries such as sequencerr, autosave, etc.
- F3RP61 can use almost all of I/O modules of FA-M3 PLC.
- F3RP61 can work with ordinary sequence CPUs which execute ladder programs. Those CPUs can communicate with each other though PLC-bus..

ISBN 978-3-95450-139-7

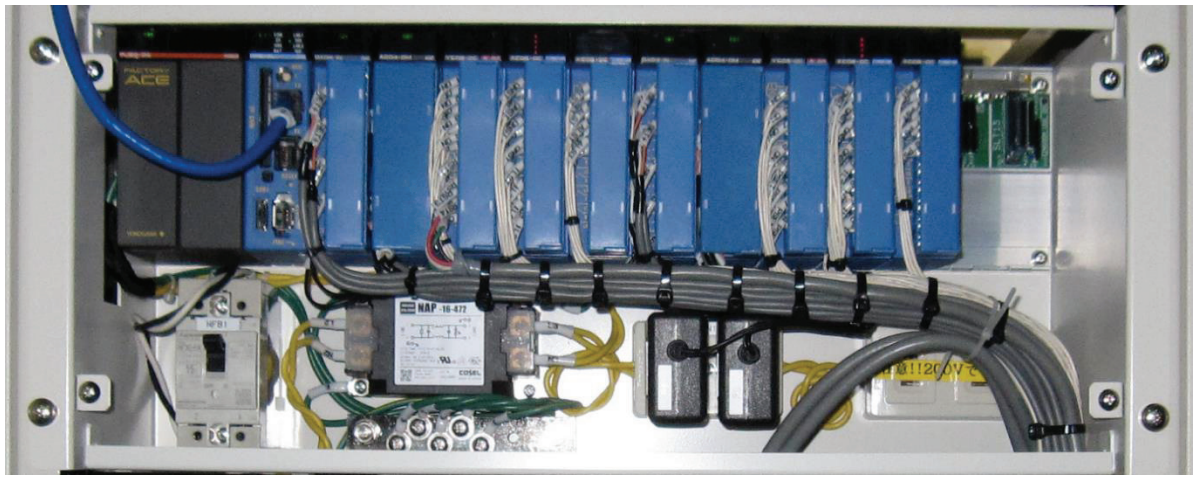

Figure 2: An example of FA-M3-based IOCs. The double-width black module is the power supply module. Next to its right is an F3RP61 CPU. All the other modulus in the figure are I/O modules.

- Real-time kernel (CONFIG_PREEMPT_RT patch [2] applied) is available as well as normal kernel.
- .Users can develop their own I/O modules under the "I/O open" license policy.

The main specifications are shown in Table 1 and an example of FA-M3-based IOC is shown in Fig. 2.

Table 1: Main Specifications of F3RP61 CPU Module

ITEM	Specification
CPU	32bit 533MHz PowerPC (Over 1000MIPS)
RAM	128MB
Flash-ROM	64MB
Interfaces	Ethernet x2, Serial I/O, IEEE1394, CF card

Executing iocCore on F3RP61

Since the F3RP61 is provided with its Board Support Package (BSP), implementing device/driver support is no more than wrapping the Application Program Interfaces (APIs) of the BSP. A RedHat-based PC is required to cross compile applications for the target (F3RP61). The BSP also includes a tool to configure the combination of prebuilt libraries which reside on the target.

Performance Measurement on I/O Access

As a fundamental performance of F3RP61, the time required to access I/O modules were measured. The measurement was done for two different cases. One measured the time in case the I/O system calls to access channel was directly invoked. The other measured the time required to process a record which was associated with the same I/O channel. Table 2. Shows the averaged time of a million times of accesses. The difference of those two is the overhead of record processing. The result indicates that the time for kernel to access the I/O module is comparable with the overhead of record processing of iocCore in user space

Table 2: Averaged Required Time (in microsecond)

I/O module type	Record Type	Record Processing	System Call
F3AD08-1R	ai	26.5	15.7
F3DA08-1X	ao	24.3	12.3
F3XD32-3F	bi	23.2	13.0

Real-time Performance of F3RP61

Another performance of importance for control applications is real-time responsiveness. In order to measure the latency, periodically executed process was used. What measured as the latency was the difference between the time when the process is scheduled to wake up and the time when the process actually woke up.
Hackbench [3] was executed as a background activity during the measurement. The result is shown in Fig. 3.

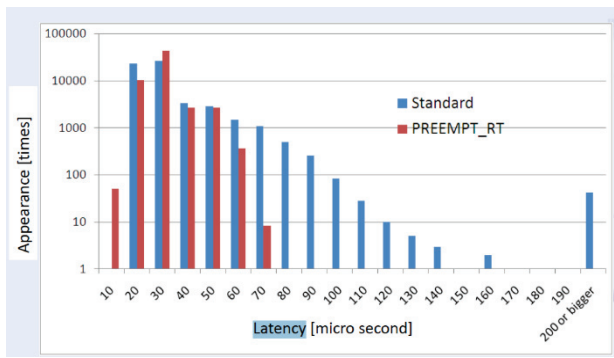

Figure 3: Result of latency measurement.

Fig. 3 shows that the worst case latency of the kernel with CONFG_PREEMPT_RT patch applied is less than a hundred micro-seconds in the measurement.
Therefore, if a system has an FPGA and it can respond to external events in tens of micro-seconds, real-time Linux

ISBN 978-3-95450-139-7

together with the FPGA can cover most of time region with regard to real-time responsiveness.

APPLICATIONS

This section describes three applications of FA-M3-based IOCs; LLRF control system, vacuum control system and status monitoring of PPS.

LLRF Control System

In order to achieve high luminosity, SuperKEKB requires better precision and stability of LLRF. In order to achieve this, a new digital LLRF system has been developed [4]. Fig. 4 shows appearance of the system. In the system, fast control, such as RF feedback and interlock upon too much RF reflection, are handled by FPGA on Advanced Mezzanine Cards (AMCs), each of which functions as an embedded IOC [5]. On the other hand, slow control is taken care of by an F3RP61-based IOC together with an ordinary sequence CPU.

The feature of the LLRF control system is that it make full use of iocCore and other related libraries, such as CA, Access Security Group (ASG), sequencer, autosave and so on.

In the LLRF system, since every AMC card and F3RP61 CPU module works as an IOC, CA functions not only as a "software bus" which interconnects remote systems but also as a "software bus" which interconnects those IOCs inside the LLRF system. For this reason, reliability of CA is crucial function for the LLRF system. So far, anything wrong has been found with CA.

ASG plays two important roles in this system. One is to realize mode switching between remote and local. The other is to protect channels, with which replaced existing internal hardwire, from write access by anything but IOCs.

Start-up sequence of the LLRF system was implemented by using EPICS sequencer. The number of states in the main state set reached about ten.

Since the LLRF control system has many parameters to set, autosave is also indispensable library. About fifty channels are being monitored, saved and restored upon reboot of the F3RP61 based IOC.

Figure 4: Digital LLRF system. FA-M3-based IOC is on the other side of the leftmost rack.

Vacuum Control System

The frontend of vacuum control system of KEKB comprised a VME-based IOC, a CAMAC crate and an FA-M3 PLC. In order to replace legacy CAMAC with newer interface, F3RP61-based IOC was introduced.

As far as digital signal concerned, it was very simple work. What needed was just to replace an unused digital I/O module of the FA-M3 PLC with an F3RP61 CPU module. Fig. 5 shows the FA-M3-based IOC under test.

Figure 5: FA-M3-based IOC for vacuum control. From left to right, a Power Supply, a normal sequence CPU an F3RP61 CPU and I/O modules.

By placing F3RP61-based IOC and ordinary sequence CPU on the same PLC-bus, the two CPUs got to be able to communicate with each other on a shared memory. It makes ladder program considerably simple to improve mainteinancability.As to digital signal, existing PLC modules can be reused. However, since the number of analogue signals to be monitored is huge, high channel density ADC was required to replace CAMAC ADCs. For this reason,

CompactRIO (cRIO), of which ADC module has thirty two channels, was adopted.

Figure 6: cRIO under installation. The light blue cable is for EtherCAT that connects main unit (upper) and extension unit (lower).

ISBN 978-3-95450-139-7

The cRIO features optional CA-Server, which allows CA clients have access to every I/O channel. As a result IOCs can communicate with cRIO without developing device/driver support.

Overload test of CA-Server on network traffic was executed by connecting a cRIO with an F3RP61 and a PC as a disturber. The result showed that CA-Server is reliable enough against bandwidth consuming traffic and rapid cycle of connection and destruction of a CA channel on the cRIO. Figure 6 shows cRIO under instllatiion.

Monitoring PPS Status

PPS was required to be reliable, solid and fail-safe. In order to meet the needs, the PPS was implemented by using only hardwire and ladder programs. EPICS-based system, on the other hand, is no more than a monitoring system. It must not be able to affect the PPS system. To be more specific, the requirement is as follows.

- Every relay status of interest must can be read by an EPICS-based monitoring system.
- Every relay must be protected from potential write access by the EPICS-based monitoring system.

In order to meet the requirement, FL-net was adopted. From programmer's point of view, FL-net configures a kind of shared memory. Each node of FL-net, can be allocated its own area (bit and/or word). Every node, including an F3RP61-based IOC, in an FL-net can read every area while any node can write only its own area. Therefore, as far as the ladder programs of PPS do not read the area which was allocated to the F3RP61-based IOC, one-sided communication was achieved. (Actually, there is no need to allocate shared memory area to F3RP61-based IOC.)

When an IOC reads bits of data, it is common practice that an IOC reads them as many as possible by using mbbiDirect-type record or waveform-type record, and cut it into bits by using soft bi-type records. In the EPICS monitoring system, however, the F3RP61-based IOC reads the relay status by bit using bi-type records for simplicity. As shown table 2, a few tens of micro-seconds suffice to process a record which access a hardware relay or register.

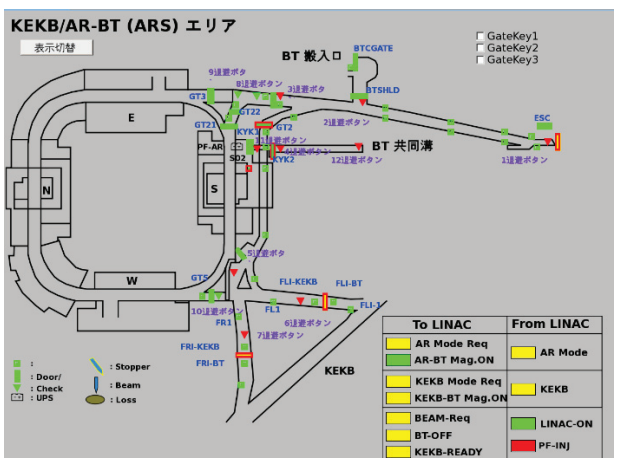

Figure 7: Display of PPS in operation.

It means that more than ten thousand of bits can be read every one second by the F3RP61-based IOC. Since the F3RP61-based IOC has nothing to do except the monitoring, resource consuming method was chosen in favour of making EPICS runtime database ultimately simple. Figure 7 shows Display of PPS in operation.

CONCLUSION

The consolidation of IOC and PLC made the frontend control layer flatter and simpler. It result in easier development and maintenance of control software. Existing VME-based IOCs and CAMAC interfaces are being replaced with F3RP61-based IOCs in control subsystems, such as LLRF, vacuum and PPS.

REMFERENCES

[1] K. Akai, "Progress in Super B-Factories" IPAC'13, Shanghi, China, May 12-17, TUYB101, (2013).
[2] https://rt.wiki.kernel.org./index.php
[3] http://people.redhat.com/mingo/cfs-scheduler/tools/hackbench.c
[4] T. Kobayashi et al., "Progress in development of New LLRF Control System for SuperKEKB" IPAC'13, Shanghai, China, May 12-17, WEPME014, (2013).
[5] J. Odagiri et al., "Fully Embedded EPICS-based Control of Low Level RF System for SuperKEKB" IPAC'10, Kyoto, Japan, May 23-28, WEPEB003, (2010).

ISBN 978-3-95450-139-7